Priscilla Hempel

CISC889010: TPCS:ADV ARTIF INTELLIGENCE

HW1

9.26.2017

**PART A**

In this section of HW1 we were asked to read in a file containing a sequence of H and P representing hydrophobic and polar amino acids forming a protein. Additionally, we were asked to build a lattice model to obtain a folded protein configuration and to calculate the energy of the configuration. In my methods, I utilize the **folding**, **calc_energy**, and **get_new_pos** functions I have written to accomplish this.

**PART B**

In this section, we were asked to implement the metropolis algorithm to simulate a population of configurations that obey the Boltzmann distribution. I accomplished this by utilizing my function **simulate**, which calls my functions **reconfig** and **calc_energy**. Below is the histogram of 1000 iterations, with temperature set to 100. Here we observe that there are only even number energies calculated. This is because in the calc_energy function I double count each topological neighbor's energy.
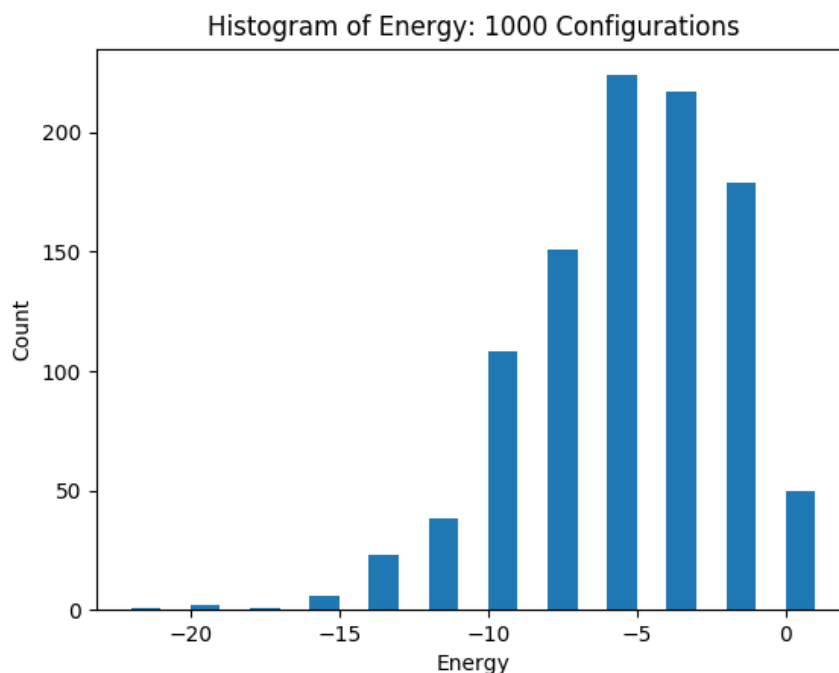
**PART C**

In this section, we were asked to optimized our configurations to have the lowest possible energy. Here, we implement the simulated annealing and the metropolis algorithm. First, I initialized a population of random configurations with size = 100. I changed the population size from very small to large and found that larger population sizes yield better optimized configurations and a lower best energy. However, code for larger population sizes take significantly longer to run.

I then chose a random member of this population and created a second generation that will be equal or smaller in size. A random member from this generation is then chosen and a new generation is generated. This process is repeated until our generation size = 1 or 0 OR the temperature has reached 0. Through all this, each legitimate child that is produced in a generation is checked whether it has a lower energy than our current lowest known configuration energy. Children are passed on to a new generation if their energy is deemed appropriate PH_HW1.py (lines 376-383). Sometimes, we allow some children with higher energies than their parent so that we may be able to overcome being in a local min of the distribution and find a lower energy. The lowest energy for a configuration that I was able to find through this method is -28. The corresponding lattice configuration can be seen on the next page.

## Optimized Configuration (protein index, lattice position)

| | | | | |
|---|---|---|---|---|
| 0,"[0, 0]" | 18,"[6, 2]" | 36,"[11, 9]" | 54,"[8, 8]" | 72,"[11, 15]" |
| 1,"[1, 0]" | 19,"[6, 3]" | 37,"[11, 10]" | 55,"[8, 7]" | 73,"[12, 15]" |
| 2,"[1, 1]" | 20,"[6, 4]" | 38,"[11, 11]" | 56,"[7, 7]" | 74,"[12, 16]" |
| 3,"[0, 1]" | 21,"[6, 5]" | 39,"[11, 12]" | 57,"[7, 8]" | 75,"[12, 17]" |
| 4,"[0, 2]" | 22,"[6, 6]" | 40,"[11, 13]" | 58,"[7, 9]" | 76,"[12, 18]" |
| 5,"[0, 3]" | 23,"[7, 6]" | 41,"[10, 13]" | 59,"[6, 9]" | 77,"[12, 19]" |
| 6,"[1, 3]" | 24,"[7, 5]" | 42,"[10, 12]" | 60,"[6, 10]" | 78,"[12, 20]" |
| 7,"[1, 2]" | 25,"[7, 4]" | 43,"[10, 11]" | 61,"[6, 11]" | 79,"[13, 20]" |
| 8,"[2, 2]" | 26,"[8, 4]" | 44,"[10, 10]" | 62,"[7, 11]" | 80,"[13, 21]" |
| 9,"[3, 2]" | 27,"[8, 5]" | 45,"[9, 10]" | 63,"[7, 12]" | 81,"[13, 22]" |
| 10,"[4, 2]" | 28,"[8, 6]" | 46,"[9, 11]" | 64,"[7, 13]" | 82,"[13, 23]" |
| 11,"[4, 1]" | 29,"[9, 6]" | 47,"[9, 12]" | 65,"[7, 14]" | 83,"[13, 24]" |
| 12,"[4, 0]" | 30,"[9, 7]" | 48,"[9, 13]" | 66,"[8, 14]" | 84,"[13, 25]" |
| 13,"[4, -1]" | 31,"[9, 8]" | 49,"[8, 13]" | 67,"[9, 14]" | 85,"[13, 26]" |
| 14,"[5, -1]" | 32,"[9, 9]" | 50,"[8, 12]" | 68,"[9, 15]" | |
| 15,"[5, 0]" | 33,"[10, 9]" | 51,"[8, 11]" | 69,"[10, 15]" | |
| 16,"[6, 0]" | 34,"[10, 8]" | 52,"[8, 10]" | 70,"[10, 14]" | |
| 17,"[6, 1]" | 35,"[11, 8]" | 53,"[8, 9]" | 71,"[11, 14]" | |